Feedback — XVII. Large Scale Machine Learning

Help

You submitted this quiz on **Mon 26 May 2014 3:05 AM IST**. You got a score of **5.00** out of **5.00**.

Question 1

Suppose you are training a logistic regression classifier using stochastic gradient descent. You find that the cost (say, $cost(\theta, (x^{(i)}, y^{(i)}))$, averaged over the last 500 examples), plotted as a function of the number of iterations, is slowly increasing over time. Which of the following changes are likely to help?

Your Answer		Score	Explanation
Try averaging the cost over a smaller number of examples (say 250 examples instead of 500) in the plot.			
Use fewer examples from your training set.			
Try using a smaller learning rate α.	✔	1.00	Such a plot indicates that the algorithm is diverging. Decreasing the learning rate α means that each iteration of stochastic gradient descent will take a smaller step, thus it will likely converge instead of diverging.
This is not an issue, as we expect this to occur with stochastic gradient descent.			
Total		1.00 / 1.00	

Question 2

Which of the following statements about stochastic gradient descent are true? Check all that apply.

Your Answer		Score	Explanation
☑ If you have a huge training set, then stochastic gradient descent may be much faster than batch gradient descent.	✔	0.25	Because stochastic gradient descent can make progress after only a few examples, it can converge much more quickly than batch gradient descent.
☐ Suppose you are using stochastic gradient descent to train a linear regression classifier. The cost function $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$ is guaranteed to decrease after every iteration of the stochastic gradient descent algorithm.	✔	0.25	Since each iteration of stochastic gradient descent takes into account only one training example, it is not guaranteed that every update lowers the cost function over the entire training set.
☐ Stochastic gradient descent is particularly well suited to problems with small training set sizes; in these problems, stochastic gradient descent is often preferred to batch gradient descent.	✔	0.25	Stochastic gradient descent is preferred when you have a large training set size; if the data set is small, then the summation over examples in batch gradient descent is not an issue.
☑ One of the advantages of stochastic gradient descent is that it can start progress in improving the parameters θ after looking at just a single training example; in contrast, batch gradient descent needs to take a pass over the entire training set before it starts to make progress in improving the parameters' values.	✔	0.25	This is true, since stochastic gradient descent updates the parameters for every training example, but batch gradient descent updates them based on an average over the entire training set.
Total		1.00 / 1.00	

Question 3

Which of the following statements about online learning are true? Check all that apply.

Your Answer		Score	Explanation
☐ When using online learning, you must save every new training example you get, as you will need to reuse past examples to re-train the model even after you get new training examples in the future.	✔	0.25	Online learning algorithms throw away old examples, incorporating them only once when they are first seen.
☐ Online learning algorithms are most appropriate when we have a fixed training set of size m that we want to train on.	✔	0.25	It is the opposite: they are most appropriate when we have a stream of training data of unbounded size.
☑ Online learning algorithms are usually best suited to problems were we have a continuous/non-stop stream of data that we want to learn from.	✔	0.25	Such a stream of data is well-suited to online learning because online learning does not save old training examples, but instead uses them once and then throws them out.
☑ When using online learning, in each step we get a new example (x, y), perform one step of (essentially stochastic gradient descent) learning on that example, and then discard that example and move on to the next.	✔	0.25	This is essentially the definition of online learning.
Total		1.00 / 1.00	

Question 4

Assuming that you have a very large training set, which of the following algorithms do you think can be parallelized using map-reduce and splitting the training set across different machines? Check all that apply.

Your Answer		Score	Explanation
☐ Logistic regression trained using stochastic gradient descent.	✔	0.25	Since stochastic gradient descent processes one example at a time and updates the parameter values after each, it cannot be easily parallelized.
☑ Linear regression trained using batch gradient descent.	✔	0.25	You can split the dataset into N smaller batches, compute the gradient for each smaller batch on one of N separate computers, and then average those gradients on a central computer to use for the gradient update.
☑ A neural network trained using batch gradient descent.	✔	0.25	You can split the dataset into N smaller batches, compute the gradient for each smaller batch on one of N separate computers, and then average those gradients on a central computer to use for the gradient update.
☐ An online learning setting, where you repeatedly get a single example (x, y), and want to learn from that single example before moving on.	✔	0.25	Since you process one example at a time, this algorithm cannot be easily parallelized.
Total		1.00 / 1.00	

Question 5

Which of the following statements about map-reduce are true? Check all that apply.

Your Answer		Score	Explanation
☑ If you have only 1 computer with 1 computing core, then map-reduce is unlikely to help.	✔	0.25	Map-reduce is a useful model for parallel computation.
☐ Running map-reduce over N computers requires that we split the training set into N^2 pieces.	✔	0.25	Usually, you will split the data into N pieces, but map-reduce does not require a specific division of the data.
☑ If you are have just 1 computer, but your computer	✔	0.25	Treating each core as a separate computer makes map-reduce just as useful with multiple

has multiple CPUs or multiple cores, then map-reduce might be a viable way to parallelize your learning algorithm.			cores as with multiple computers.
Because of network latency and other overhead associated with map-reduce, if we run map-reduce using N computers, we might get less than an N-fold speedup compared to using 1 computer.	✔	0.25	The maximum speedup possible is N-fold, and it is unlikely you will get an N-fold speedup because of the overhead.
Total		1.00 / 1.00	